

Ahmed Sherko

Responsiivisuuden huomioiminen verkkosivujen suunnittelussa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

25.11.2014

Tekijä	Ahmed Sherko
Otsikko	Responsiivisuuden huomioiminen verkkosuunnittelussa
Sivumäärä Aika	26 sivua 25.11.2012
Tutkinto	insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja	yliopettaja Kari Järvi
<p>Tämän insinöörityön aiheena oli responsiivisuuden huomioiminen verkkosuunnittelussa. Responsiivinen verkkosivu on sellainen, joka mukautuu itsestään, kun laitteen näytön koko ja resoluutio muuttuvat. Ihmiset voivat näin selata internetsivuja millä tahansa laitteella, jossa internetin käyttö on mahdollista.</p> <p>Tavoitteena oli selvittää tärkeät responsiivisen verkkosivun käytettävyyteen vaikuttavat tekijät, jotta ihmisille luotaisiin loistava kokemus heidän käyttämästä laitetypistä tai -koosta riippumatta.</p> <p>Tekstin tarkoituksena oli esittää responsiivisen verkkosuunnittelun peruseräatteen ja toteutumistavat käytettävyyden osa-alueena selkeästi ja yksinkertaisesti tietotekniikan alan opiskelijoille, verkkosivun suunnittelijoille ja muille aiheesta kiinnostuneille. Tultiin lopussa siihen päätökseen verkkosivuja kannattaa rakentaa responsiivisesti, jottei eri resoluution omaaville laitteille tarvitsisi erikseen luoda eri ohjelmakoodia. Tämä lisäisi kustannuksia, edellyttäisi enemmän henkilöstöä ja kuluttaisi aikaa. Verkkopalvelun responsiivisessa suunnittelussa on tärkeä ottaa käyttäjien käyttökokemusta huomioon. Hyvä verkkopalvelu on nopea, selkeä, yksinkertainen, helppo oppia ja virheetön.</p> <p>Tekstissä esitellään myös Mobile Firstin käsitettä. Mobile Firstin mukaan verkkosivun suunnittelu olisi helpompaa ja yksinkertaisempaa tehdä ensin pienikokoisille mobiililaitteille, ja vasta sitten tietokoneille. Koska mobiilikäyttäjiä on nykyään runsaasti, verkkosivun suunnittelusta mobiililaitteille tulee todennäköisesti joka tapauksessa tarpeellista.</p>	
Avainsanat	responsiivisuus, verkkosivu, ohjelmointi, internet, Mobile First, käytettävyys

Author Title	Ahmed Sherko Considering usability in creating responsive websites
Number of Pages Date	26 pages 25 November 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software engineering
Instructor	Kari Järvi, Principal Lecturer
<p>The objective of this thesis is to encourage programmers to give higher regard to usability when creating responsive websites. A responsive website is one which automatically adapts to the resolution of a device. People can thus enter a website through any device they wish. It is increasingly important to build websites responsively, as nowadays so many different-sized devices exist that building a different website for each one would be relatively time-consuming and expensive. Additionally, more staff would be needed to create and maintain these sites.</p> <p>In this thesis the concept of Mobile First is presented. According to Mobile First the design of a website would be remarkably easier and simpler if the design was first created for a mobile device, and afterwards for computers. Because so many mobile users exist today, designing websites for mobile devices will probably be necessary anyway.</p> <p>This thesis is mainly based on a literature review. The internet and texts written by experts in this field have been useful. The aim of this thesis is to describe the basic principles of responsive web design to students of information technology, programmers and others interested in the field. The objective of this thesis is to investigate the factors that affect the usability of responsively built websites. A website should offer excellent experience no matter which device type or size the user chooses to access the internet from.</p>	
Keywords	responsiveness, website, programming, internet, Mobile First, usability

Sisällys

1	Johdanto	1
2	Responsiivisuuden kehitys	2
2.1	Mobiililaitteiden historiaa	4
2.2	Mobile First	6
3	Verkkosivujen toiminta ja ulkoasu	9
4	Responsiivisten verkkosivujen suunnittelun periaatteet ja toteuttamistavat	11
4.1	Joustava sommittelu	12
4.2	Joustavat kuvat	13
4.3	Viewport-määrittely	14
4.4	Mediakyselyjen käyttö	15
4.5	Keskeytyskohdat	19
4.6	Suunnittelukehys	20
4.6.1	Suunnittelukehysten käytön edut	20
4.6.2	Sopivan suunnittelukehysten valinta	21
4.6.3	Suunnittelukehysten haittapuolet	21
5	Käytettävyys	23
5.1	Navigointi	23
5.2	Painikkeiden suunnittelu.	24
5.3	Verkkosivun vierittäminen	24
5.4	Hakutyökalut	24
6	Yhteenveto	25
	Lähteet	27

Lyhenteet ja käsitteet

Breakpoint	Breakpoint on piste, joka tarjoaa sivustolle parhaan mahdollisen ulkoasun esittämään informaatiot.
CSS	Cascading Style Sheets on www-dokumenteille kehitetty tyyliohje.
HTML	Hypertext Markup Language, WWW-sivujen koodauskieli.
HTTP	Hypertext Transfer Protocol on verkkoprotokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoa varten.
HTTPS	Hypertext Transfer Protocol Secure on HTTP- ja SSL/TLS-protokollan yhdistelmä, jota käytetään salatun tiedon tiedonsiirtoon.
IE	Internet Explorer, Microsoftin internetselainohjelma.
Mobile First	Termiä Mobile First käytetään, kun WWW-sivun koodausta aloitetaan ensin mobiileista.
OS-päivitys	Käyttöliittymän päivitys.
Px	Pikseli (engl. pixel) on kuvapiste.
Resoluutio	Sivun tai kuvan erottelutarkkuus.
Responsiivinen	Sana responsiivinen on johdettu englanninkielisestä sanasta responsive, joka tarkoittaa, että sivusto skaalautuu sopivasti käyttäjän näytön koon mukaan.

1 Johdanto

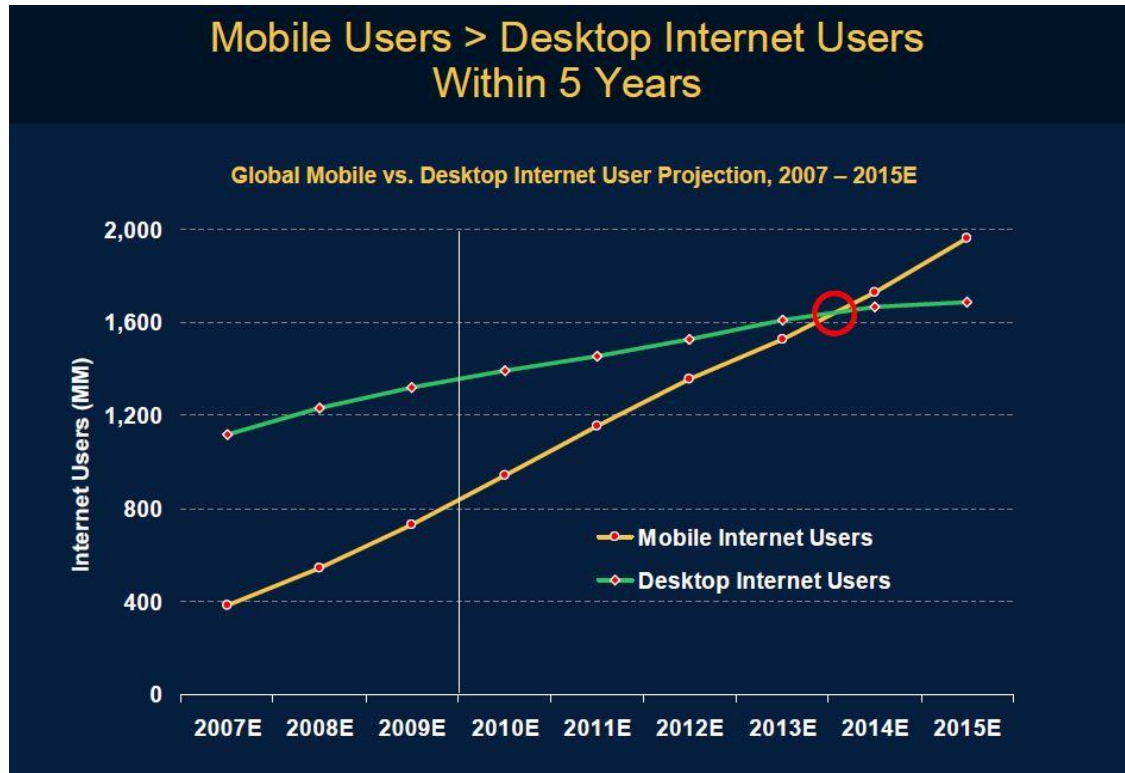
Tämän insinööriyön tarkoituksena on selvittää verkkosivujen suunnitteluun vaikuttavia tekijöitä, kun sivun tulee soveltua useaan eri näyttökokoon. Työssä käsitellään joitakin esimerkkejä mukautuvasta, responsiivisesta verkkosuunnittelusta. Tarkoituksena on esitellä menetelmiä, jotka vaikuttavat vahvasti, kun aiotaan tehdä uuttaa käyttöliittymää. Tämä insinööriyö on tärkeä, koska se kuvaa miksi koodaamiseen kannattaisi perehtyä alkuvaiheessa, että myöhemmin välttyisi vaikeuksilta.

Insinööriyössä esitetään kysymyksiä, joihin yritetään vastata myöhemmin. Olisiko helpompi ja kannattavampi aloittaa sivuston koodaus mobiileista vai pöytätietokoneista? Mitä eroa niillä on? Mitä ongelmia kussakin vaihtoehdossa on? Työssä testataan, mitä vaikutuksia käyttäjän näkökulmasta, ja mitä responsiivisuus tuo mobiili- ja pöytätietokoneen käyttäjälle. Testauksessa otetaan huomioon navigointi, painikkeiden suunnittelu ja verkkosivun vierittäminen. Insinööriyö perustuu pääasiassa kirjoista hankittuun tietoon. Internet ja alan asiantuntijoiden tekstit ovat olleet hyödyllisiä.

Insinööriyön toisessa luvussa kerrotaan responsiivisuuden kehityksestä ja mobiililaitteiden historiasta. Kolmannessa luvussa esitetään verkkosivun rakenne ja toiminta sekä joustavan sommittelun idea yleisellä tasolla. Neljännessä luvussa käsitellään joustava sommittelu, joustavat kuvat, mediakyselyt ja niiden käyttötavat, keskeytyskohdat ja suunnittelukehysten käyttö. Viidennessä luvussa esitetään lyhyesti erilaisia menetelmiä, joilla verkkosivun käytettävyyttä voidaan kohentaa [1.]

2 Responsiivisuuden kehitys

Internetin käyttäminen on nykyään mahdollista monilla eri laitteilla. Vuonna 2014 ensimmäisen kerran historiassa suurempi osa ihmisistä suosii internetin käyttöä mobiililaitteella tietokoneen sijasta (kuva 1).



Kuva 1. Vuonna 2014 ihmiset käyttävät mobiililaitteita enemmän kuin tietokoneita internetin käyttöön. [11.]

Kuluttajilla on nyt valinnanvaraa, käyttääkö internetiä mobiililla, tabletilla vai tietokoneella. Tableteilla on sekä mobiilien että tietokoneiden etuja. Kevyen painonsa vuoksi niitä voidaan käyttää melkein missä vain. Toisaalta tietokoneen tavoin tabletin näyttö on melko iso. Eri laitteilla on erikokoisia näyttöjä. Mukavan internetkäytön ansiosta mobiililaitteiden käyttäjät ovat lisääntyneet eksponentiaalisesti viime vuosina, ja tämän kasvun oletetaan jatkuvan tulevaisuudessa. Pienikokoiset laitteet ovat nyt suosittuja laitteita internetin selailuun käytettäväksi, sillä mobiililaitteet voivat nyt esittää internetsivuja yhtä hyvin kuin tietokoneet. [1;4.]

Verkkosivujen tarkasteluun käytetään tavallisesti verkkoselainohjelmaa. Käyttäjä lähettää verkkosivun kautta palvelimelle pyynnön esimerkiksi klikkaamalla verkkosivulla

olevaa linkkiä ja palvelin vastaa pyyntöön lähettämällä käyttäjän verkkoselainohjelmalle pyydetyn sisällön tarkasteltavaksi. Sisältö voi koostua tekstistä, kuvista, videosta tai muusta informaatiosta. Verkkoselainohjelman vastaanottama tieto muotoillaan käyttäjälle näytettäväksi verkkosivuksi esimerkiksi HTML- ja CSS-kielillä.

HTML (Hypertext Markup Language) asettaa sisällölle rakenteen ja sen roolin dokumentissa. CSS:n avulla dokumenttiin saadaan haluttu tyyli, kuten esimerkiksi värimaailma ja sommittelu. CSS-tyylit voidaan kirjoittaa suoraan dokumenttiin, mutta yleensä ne sijoitetaan erilliseen tiedostoon, jolloin CSS-tiedostoihin annetaan viittaus HTML-dokumentissa. HTML:n ja CSS:n avulla pystytään toteuttamaan suurin osan verkkosivulle sijoitettavista ominaisuuksista, mutta joskus tulee vastaan tapauksia, joihin nämä tekniikat eivät riitä. JavaScriptiä tarvitaan esimerkiksi tiettyihin navigaatoratkaisuihin ja joidenkin mobiilivalikoiden toteuttamiseen.

Kun verkkosivujen käyttö alkoi yleistyä, niiden tarkasteluun sopivat päätelaitteet olivat työpöytäkoneita, jotka ovat toiminnaltaan ja ominaisuuksiltaan samankaltaisia. Verkkosivut suunniteltiin toimimaan tässä ympäristössä ja niiden käyttö perustui näppäimistöön ja hiireen.

Kun mobiililaitteiden käyttö verkkosivujen selailussa yleistyi, ensimmäinen tapa vastata kysyntään oli rakentaa täysin erilliset mobiilisivut. Nykyaikana erikokoisia laitteita on kuitenkin niin paljon, että eri koodia ei eri laitteille kannata luoda, vaan sivu on helpompi tehdä *responsiiviseksi* siten, että samalla koodilla sivu skaalautuu erikokoisille näytöille sopivaksi. Tulevaisuudessa uusia laitteita luodaan yhä lisää, minkä vuoksi eri koodin luominen erikokoisille näytöille ei ole suositeltavaa. Esimerkiksi Microsoft Omnipotent -prototyyppi voi muuntaa minkä tahansa pinnan näytöksi, jopa käden. Tällöin, jos sivustoa ei ole suunniteltu responsiivisesti, jokaiselle eri resoluution omaavalle näytölle pitäisi muodostaa oma koodi, kuten monet suunnittelijat tänä päivänä tekevät. Usein verkkosivun suunnittelijat kehittävät yhden sivun tietokoneille ja toisen mobiililaitteille. Joillakin yrityksillä on eri verkkosivu tietokoneille, tableteille, kosketusnäytöillä mobiililaitteille sekä muille mobiililaitteille. Tämä tarkoittaisi neljää sivua, jotka tarvitsevat päivittämistä, testaamista ja ylläpitoa. Responsiivinen verkkosuunnittelu eliminoisi tarpeen luoda eri ohjelmakoodia aina kun uusi laite ilmestyy markkinoille.

Toukokuussa 2010 Ethan Marcotte kirjoitti artikkelin nimeltä "Responsive Web Design" A List Apart -nimiselle verkkolehdele. Marcotte näytti, että yksi ainoa koodiperusta voi-

si toimia monilla eri resoluutioilla, jos sivu on suunniteltu responsiivisesti. Responsiivisen verkkosivun suunnittelu on lähestymistapa, jonka tarkoituksena on suunnitella verkkosivuja niin, että ne mukautuvat vastaamaan laitteen näytön resoluutioon ja orientaatioon. Toisin sanoen, kun jotakin verkkosivua selataan erikokoisista laitteista, sivun kuvien koot, teksti, ja muu sisältö mukautuvat itsestään muuttuneelle resoluutiolle. Tällöin eri koodia ei tarvitsisi luoda erikseen eri laitteille ja rahaa säästyisi. Lisäksi enemmän suunnittelijoita tarvittaisiin sivujen ylläpitoon, sillä pelkästään yksi sivu tarvitsee monta suunnittelijaa sen ylläpitoon. [5.]

Mobiililaitteiden lisääntyessä sellaisten verkkosivujen luominen, jotka ovat nähtävissä puhelimessa ja muissa mobiililaitteista, on tullut välttämättömäksi suunnittelijoille. Responsiivisessa verkkosuunnittelussa on tärkeä pohtia sivulla käyvien vierailijoiden käyttökokemusta. Toimivan verkkopalvelun lisäksi tavoitteena on tarjota käyttäjille mahdollisimman mukava kokemus heidän käyttämästään elektronisesta laitteesta ja valitseman selaimesta riippumatta. Tämän työn tavoitteena on korostaa käyttäjien näkökulman pohtimista responsiivisessa verkkosuunnittelussa. Navigaation vaikea löytäminen, linkkien hankala napsauttelu tai yli 20 sekunnin sivun latautuminen saatavat johtaa vierailijoita hylkäämään sivuston, ja hakemaan parempaa verkkopalvelua toisilta sivustoilta.

Marcotten esityksen mukaan verkkosivuista tehdään mukautuvia käyttämällä niin kutsuttua joustavaa asemointia, CSS3-mediakyselyjä sekä joustavia kuvia.

2.1 Mobiililaitteiden historiaa

Elektroniset laitteet ovat kehittyneet ajan myötä valtavasti. Tietoverkot olivat aluksi hyvin hitaita, tietokoneiden näyttöjen koot pieniä ja syöttömenetelmät ärsyttäviä. Vuonna 1984 julkaistiin ensimmäinen Macintosh-tietokone 512 x 342 pikselin resoluution näytöllä. Ajan myötä tietokoneiden näyttöjen resoluutio kasvoi. Kymmenen vuotta myöhemmin julkaistiin Apple Multiple Scan 1024 x 769 pikselin resoluution näytöllä. Teknologian kehittyessä tietokoneet ja mobiililaitteet kehittyivät. Esimerkiksi nykyisin perinteisen hiiren lisäksi kosketuslevy on toinen suosittu osoitinlaite. [3.]

Alkuperäinen iPhone, joka mahdollisti yhtäkkiä internetin käytön myös mobiililaitteella, tuli markkinoille vuonna 2007. iPhone, pienellä 320 x 480 pikselin resoluution näytöllä,

toi pienikokoiset laitteet suosioon. Tarjottuaan mukavaa internetin käyttöä iPhonesta ja toisista älypuhelimista tuli varsin suosittuja ihmisten keskuudessa. Tabletit, kuten Applen iPad, omaavat tietokoneen tavoin ison näytön. Toisaalta kevyen painonsa ansiosta niitä voi käyttää melkein missä vain, bussissa, sohvalla tai bussipysäkillä. Pelikonsolit, kuten Microsoft Xbox 360 ja Nintendo Wii, ilmestyivät sisäänrakennettujen selaimien kanssa sallien käyttäjiään esittämään verkkosivuja televisionäytöillä. E-kirjojen lukulaitteissa, kuten Amazonin Kindle-laitteissa Nook Barnesissa ja Noblessa, on myös verkkoselain.



Kuva 2. Uusia laitteita tuotetaan jatkuvasti. [12.]

Nykyisin suosittujen laitteiden näytöt vaihtelevat 280 pikselin leveydestä 1920 pikselin leveyteen (kuva 2). Elektronisten laitteiden monipuolisuus (taulukko 1) selittää, miksi verkkosivujen pitäisi olla käytettävissä kaikenkokoisilla näytöillä. [1;3.]

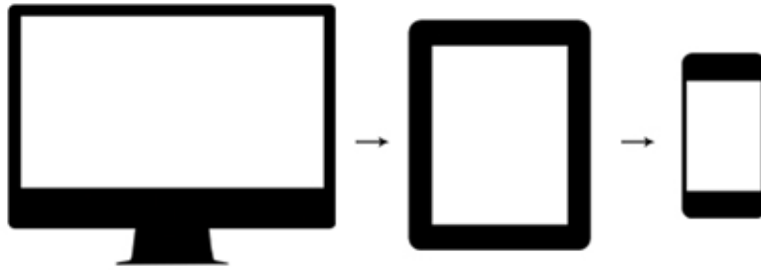
Taulukko 1. Erikokoisia laitteita on nykyään niin paljon, että jokaiselle resoluutiolle ei kannata lähteä rakentamaan eri verkkosivua. [12]

320 pixels	For small screen devices, like phones, held in portrait mode.
480 pixels	For small screen devices, like phones, held in landscape mode.
600 pixels	Smaller tablets, like the Amazon Kindle (600×800) and Barnes & Noble Nook (600×1024), held in portrait mode.
768 pixels	Ten-inch tablets like the iPad (768×1024) held in portrait mode.
1024 pixels	Tablets like the iPad (1024×768) held in landscape mode, as well as certain laptop, netbook, and desktop displays.
1200 pixels	For widescreen displays, primarily laptop and desktop browsers.

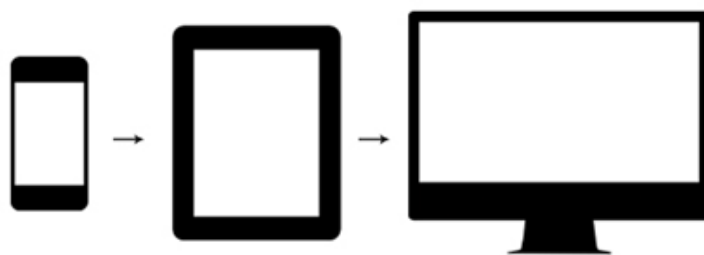
2.2 Mobile First

Mobile First -käsitteen kehitti Luke Wroblewski vuonna 2009. Mobile Firstillä tarkoitetaan sitä, että verkkosivu suunnitellaan ensiksi mobiililaitteelle ja muokkauksia tehdään sen jälkeen tietokoneelle suunnatulle versiolle. Termin suomennos on mobiililähtöinen suunnittelu. [2.]

Iso virhe, jonka verkkosuunnittelijat usein tekevät, on verkkosivun suunnitteleminen ensin pöytätietokoneelle ja sitten sen muuttaminen pienikokoisemmalle näytölle sopivaksi puristamalla, piilottamalla ja jälleenjärjestämällä sisältöä (ks. kuva 3). Tätä latis-tamisen paradigmaa (graceful degradation) käytettäessä sivut näyttävät hyviltä mobiililaitteissakin, mutta lopputuloksena on yleensä heikosti toimiva ja sekava lopputulos. [2.]



Kuva 3. Responsiivisen verkkosuunnittelun aloittaminen tietokoneesta. [2.]



Kuva 4. Mobile First -periaate [2.]

Kuten nimi viittaa, Mobile First on ajattelu, jonka mukaan suunnittelu aloitetaan pienemmästä laitteesta isompaan. Tämä varmistaa sen, että sisältö tulee skaalautumaan oikein (ks. kuva 4).

Hyvä ja käytettävä verkkosivu on sellainen, joka ottaa asiakkaan tarpeet huomioon. Sivuston käyttäjät tahtovat tärkeää ja oleellista informaatiota nopeasti ja helposti. Liian monet suunnittelijat suunnittelevat verkkosivua mobiililaitteelle pöytätietokoneelle räätälöidyn sivun perusteella. Tällöin he valikoivat sisältöä pöytätietokoneen sivuista mobiililaitteille suunnitetuille sivuille.

Mobiililaitteella on vähemmän tilaa sisällölle, mikä auttaa luopumaan kaikesta ylimääräisestä ja epäoleellisesta sisällöstä, joka mahtuisi isommalle näytölle. Siten on pakko miettiä, mitkä ominaisuudet, toiminnot ja sisällöt ovat oikeasti tärkeitä sisällyttämään pienen 320 x 480 pikselin näytön ruudulle. Sivun suunnittelun aloittaminen mobiililaitteesta auttaa välttämään turhan sisällön sisällyttämistä pöytätietokoneen versiolle. Se mikä ei sovi mobiiliin näytölle, ei ehkä ole tärkeä isommallekaan näytölle. Kun verkkosi-

vulle on vähemmän sisältöä laitettavaksi, myös sivun suunnittelijoiden aikaa säästyy.
[1;3.]

Tällöin olennaista sisältöä ei tule karsittua. Koska mobiilikäyttäjien määrä on kasvussa, yritysten kannattaa panostaa hyvän käyttökokemuksen tarjoamiseen, jotta vierailijat palaisivat sivulle uudelleen. On ennustettu, että vuonna 2020 mobiilikäyttäjiä tulee olemaan 2,2 miljardia. Yli 20 prosenttia yhdysvaltalaisista ja englantilaisista käyttää internetiä pelkästään mobiililaitteellaan. Koska internetin käyttämisen mobiililaitteella ennustetaan lisääntyvän tulevaisuudessa, on tärkeämpi miettiä mobiililaitteen sisältöä ensin.
[1;3.]

3 Verkkosivujen toiminta ja ulkoasu

Useimpien verkkosivujen sommittelurakenteet koostuvat ylätunnisteesta (header), alatunnisteesta (footer) sekä navigointi- ja sisältöelementeistä. Verkkosivujen ulkoasuja on erilaisia. Kiinteät ulkoasut (fixed-width layout) ovat yleisimmin käytössä. Sivun leveyden määrittää tietty pikseliarvo (pikselien lukumäärä vaakasuunnassa). 960 pikseliä on yleisimmin käytetty nykyään. Tämän ulkoasun ongelmat tulevat esille, kun käyttäjä vaihtaa laitteen ruudun kokoa. Jos sivu on 960 pikseliä leveä ja sivua katsotaan pienemmästä näytöstä, vierailija näkee vain osan sivusta ja ruma vaakatasoinen vierityspalkki esiintyy sivulla. Isoilla näytöillä sen sijaan iso valkoinen alue ympäröi sivua. Koska eri laitteilla on erikokoisia näyttöjä ja pikselitiheyksiä, pikseleillä mitattu sisältö, kuten teksti, joka näyttää hyvältä yhdessä laitteessa, saattaa näyttää huonolta toisessa. [3.]

Taitolla tarkoitetaan verkkosivun elementtien sommittelua ja tyyllittelyä haluttuun muotoon. Sivua voidaan jakaa pystysuoriin palstoihin. Tämän lisäksi sivu usein jaetaan myös vaakasuuntaisiin osiin joko ylä- tai alareunasta tai molemmista. Keskelle jäävä tila voidaan lisäksi jakaa kahteen tai kolmeen palstaan. Taittopohja eli gridi (grid) on apuruudukko, jota käytetään verkkosivujen asemointiin. Sen ideana on sivun jakaminen määrätyn levyisiin palkkeihin. Palkkeja apuna käyttäen kootaan sivun rakenteen perusta ja elementit sijoitellaan sivulle.

Mukautuva suunnittelu mahdollistaa sen, että verkkosivusta tehdään yksi versio, joka skaalautuu sopivaan kokoon eri päätelaitteilla katsottaessa. Tällöin katsojalle näytetään eri päätelaitteelle optimoitua sisältöä muodossa, joka toimii eri näytön leveyksillä. Isolla näytöllä voi esimerkiksi olla kolme palsta rinnakkain, tabletilla kaksi ja matkapuhelimella. Mukautuvan verkkosivun suunnittelussa hyödynnetään kolmea perustekniikkaa, joita sovelletaan joustavan sommittelun lisäksi.

Joustavassa asemoinnissa (joustava sommittelu, fluid grids) käytetään suhteellisia mittoja suhteessa käytettävissä olevaan alaan ja leveydet merkitään pikselien asemasta prosentteina koko leveydestä. Siten verkkosivuista tulee joustavimpia selaimen ikkunan ja näytön resoluution suhteen.

Näyttöruutu on selaimen näkyvä alue eli selaimen leveys. Esimerkiksi määritellään sivu, jossa pääsarake on 60 % leveä näyttöruudun leveyden suhteessa, oikealla sivu-

palkilla on sarake joka käyttää 30 % näyttöruudun leveydestä ja muu rakenneosaa on 10 % leveä. Nyt ei ole väliä, selaako käyttäjä sivua 1024 pikselin pöytätietokoneen selaimesta vai 768 pikselin tabletista, sillä elementtien leveydet mukautuvat automaattisesti ruudulle sopivaksi niin, että suhteet pysyvät muuttumattomina.

Kuvat asetetaan skaalautuviksi (flexible images) ja elementtien leveydet määritellään prosentteina. Mediakyselyt (media queries) on tekniikka, jolla laitteet jaetaan eri ryhmiin, joille tehdään erilaiset asetukset.

Taittopohjan (gridi) käyttöönottamisella saadaan tasapainoa, välistystä ja järjestystä sivustoon, ja luettavuus ja silmäiltävyys kehittyvät. Lisäksi se lisää luovuutta, auttaa käyttäjiä tietämään, mistä löytää informaatiota ja helpottaa uuden sisällön lisäämistä. Gridin käyttö siis parantaa sivun sisällön rakennetta ja johdonmukaisuutta. [3.]

Mediakyselyt ovat tärkein työkalu responsiivista internetsivua tehtäessä. Niiden avulla päätelaite ja selain ymmärtävät, mitä tyyliä niiden pitää milloinkin käyttää.

Joustavalla asemoinnilla ehkäistään kiinteän ulkoasun ongelmat. Vaakatasoisesta vierityspalkista sekä valkoisesta tilasta päästään eroon. Ulkoasun muokkaaminen mediakyselyjen avulla ja tyylien soveltaminen on myös helpompaa joustavassa asemoinnissa. [3.]

4 Responsiivisten verkkosivujen suunnittelun periaatteet ja toteuttamistavat

Responsiivisten verkkosivujen suunnittelun tarkoituksena on saada verkkosivu mukautumaan käyttäjän laitteen näytön kokoon ja orientaatioon. Tällöin katsojalle näytetään eri päätelaitteelle optimoitua sisältöä muodossa, joka toimii eri näytön leveyksillä. Responsiivisuus sallii verkkosivun mukautumisen pienemmälle tai suuremmalle näyttöruudulle. Samaa sivua voidaan siis käyttää kaikilla laitetyypeillä, ilman että tarvitaan useita erillisiä verkko-osoitteita saman sisällön jakamiseen.

Suunnitteluprosessin päävaiheita ovat mukautuvan verkkosivuston lisävaatimusten selvittäminen ja sivujen sisällön kartoittaminen, sisällön kuten kuvien ja tekstin määrittely ja priorisointi eri laitteille, layoutin suunnittelu eri resoluutioille, joustava asemointi, joustava kuva ja media, sekä CSS3-mediakyselyiden käyttö. Tässä luvussa käsitellään responsiivisen verkkosuunnittelun periaatteet ja käyttötavat sekä näyttöruutu ja suunnittelukehykset. [1.]

Responsiivisen verkkosivun rakentamista varten tarvitaan verkkoselaimia, koodieditoita sekä internetin käytön mahdollistavia laitteita. Tämän opinnäytetyön lukijan oletetaan jo hallitsevan perustiedot HTML:n, CSS:n ja JavaScriptin käytöstä, joten näiden ohjelmien alkeita ei tässä tekstissä esitellä. Tässä luvussa eri verkkoselaimia, joita voi ladata ilmaiseksi internetistä, tarvitaan kehitettävien responsiivisten verkkosivujen testaamiseksi näillä selaimilla. Suosituihin verkkoselaimiin kuuluvat Firefox, Chrome, Opera, Safari ja Internet Explorer. Koodieditori, kuten esimerkiksi Sublimetext, on välttämätön työkalu responsiivisessa verkkosuunnittelussa. Taulukko 2 antaa esimerkkejä joistakin koodieditoreista, jotka toimivat tietyillä käyttöliittymillä.

Taulukko 2. Esimerkkejä joistakin koodieditoreista, jotka toimivat eri käyttöliittymillä. [12.]

OS	Code editors
Windows	Notepad++ (http://notepad-plus-plus.org/)
	WebMatrix (http://www.microsoft.com/web/webmatrix/)
	TextPad (http://www.textpad.com/)
OS X	TextWrangler (http://www.barebones.com/products/textwrangler/)
	MacVim (http://code.google.com/p/macvim/)
	Brackets (http://brackets.io/)
Linux	Gedit (http://projects.gnome.org/gedit/)
	Geany (http://www.geany.org/)
	BlueFish (http://bluefish.openoffice.nl/index.html)

Jotta suunnittelija pystyisi näkemään verkkosivunsa erikokoisista näytöistä, monen laitteen ostamisen sijasta suositeltavampaa on käyttää responsiivista sovelluskirjanmerkkiä (bookmarklet), sillä se on taloudellisempi ja yksinkertaisempi vaihtoehto. Sovelluskirjanmerkki (kirjanmerkkisovelma) on selaimen kirjanmerkkeihin tallennettava pieni sovelma, jonka toiminta on tallennettuna kirjanmerkin URI-osoitteeseen. Sovelluskirjanmerkkiä käytetään tavallisen kirjanmerkin tapaan ja se on yksinkertaisesti toimintoja selaimen lisäävä, yhden napsautuksen työkalu. Sovelluskirjanmerkki (taulukko 3) on sisäänrakennettu ominaisuus Firefox 15:ssä ja sen jälkeisissä versioissa. Chromella on tarjolla samantyyppinen Windows Resizer -niminen laajennus [4.].

Taulukko 3. Erilaisia sovelluskirjanmerkkejä, joilla voi testata verkkosivuja erikokoisilla näytöillä. [12.]

RWD demonstration (<http://jamus.co.uk/demos/rwd-demonstrations/>)
 Screenqueri.es (<http://screenqueri.es/>)
 Responsinator (<http://www.responsinator.com/>)
 ResponsivePX (<http://responsivepx.com/>)
 Resizer (<http://codebomber.com/jquery/resizer/>)
 Screen Fly (<http://quirktools.com/screenfly/>)
 Adobe Edge Inspect (<http://html.adobe.com/edge/inspect/>)

4.1 Joustava sommittelu

Joustavassa sommittelussa (selainikkunan koon mukana joustava sivuston asettelumalli) elementit ilmaistaan prosentteina koko alueesta muuttumattomien ja epäjoustavien pikseleiden sijasta. Menettelystä käytetään nimityksiä gridi, ristikko, pohjaverkko ja asettelupohja. Joustavassa sommittelussa (flexible grid-based layout, fluid grids) ilmaistaan pisteverkon jokainen komponentti, mukaan lukien rivit/sarakkeet suhteellisina koko alueeseen nähden.

Suunnittelun aluksi määritellään suurin näyttökoko. Sen jälkeen ristikko jaetaan sarakkeisiin. Lopuksi määritellään kunkin elementin suhteellinen leveys ja korkeus suurimpaan näyttökokoon verrattuna. Näin luodaan sommittelu, joka mukautuu selaimen ikkunan koon mukaan.

Jos esimerkiksi koko leveys on 960 pikseliä ja sisältökenttä 880 pikseliä, reunojen leveydeksi jää 40 pikseliä. Sisältökentässä on kaksi saraketta leveydeltään 640 ja 220

pikseliä. Joustavassa suunnittelussa leveyttä 960 pikseliä merkitään arvolla 100 %, jolloin sisältökentän kooksi tulee $880/960 = 0,917$ eli 92 %. Sisältökentän sarakkeiden kooksi saadaan vastaavasti 72 % ja 25 %.

Suurella näytöllä sivun sisältö keskitetään koko ruudulle siten, että vasemmalla ja oikealla ei ole oikeastaan mitään sisältöä ja koko sisältö näkyy kerralla kokonaan ja sijoittuu koko otsikkoon ja koko alla olevaan rakenteeseen eri levyisine osioineen. Keskikoossa sisällön leveys pienenee, eikä koko sisältö enää mahdu kerralla ruudulle, mutta rakenne säilyy samanlaisena kuin isommillakin näytöillä. Pienimmässä näyttökoossa koko sisältö asettuu yhteen sarakkeeseen rullattavaksi pystysuunnassa.

Joustavan sommittelun toteutus ilman vaatii sivun rakentajalta paljon työtä. Suositeltava tapa on käyttää jotain CSS grid -generaattoria (esimerkiksi Variable Grid System) tai tukeutua olemassa olevaan valmiiseen pohjaan. [1.]

4.2 Joustavat kuvat

Tyypillisesti verkkosivulla olevan kuvan koko määritellään tarkasti pikseleinä. Tämä tapa toimii, jos verkkosivun asettelun mittasuhteet pysyvät aina samoina. Kuitenkin erilaisiin tilanteisiin mukautuvia verkkosivuja suunnitellessa kuvien koon tulisi muuttua muun sisällön mukana. Jotta kuva skaalautuisi laitteen näyttöruudulle, kuvaelementille määritellään CSS-tyyppisääntö:

```
img {
    max-width: 100%;
}
```

Määrittelyn seurauksena kuvan leveys ei saa ylittää selaimen ikkunan leveyttä. Kuvan koko muuttuu käyttäen prosenttimäärää ja mukautuu selaimen leveyden muutoksen mukaisesti. Olisi kuitenkin huomattava, että jotkut selaimet, kuten IE6 ja alemmat versiot, käyttävät edellisen koodin sijasta "width: 100" -merkintää, joka samoin estää kuvan ylittämästä ikkunan leveyttä. Max width -merkinnän ansiosta, sivuston kuvat mukautuvat responsiivisesti joustavan ulkoasun mukaisesti. Kuva voi nyt näkyä hyvin monien laitteiden näytöillä. Samalla merkinnällä myös videoista ja muusta mediasta saa mukautuvan. [1;2.]

```
embed, object, video {
    max-width: 100%;
}
```

Toinen hyödyllinen tapa saada kuvia joustavalle sommittelulle on ylityksen käyttäminen. Overflow-määrittelyllä kerrotaan, mitä tehdään näyttöalueen ulkopuolelle jäävälle osalle. Esimerkiksi `overflow: hidden` leikkaa elementtiruudun yli menevän leveän kuvan osan. Kuva on edelleen paikallaan, mutta ylimääräiset palat on piilotettu.

```
div {
    overflow: hidden;
}
```

Muita komennon käyttömuotoja ovat oletuskäyttäytyminen `visible`, jossa yli menevä osa jää näkyviin ruudun ulkopuolelle ja `scroll`, jossa vierityspalkki tulee näkyviin. [4.]

4.3 Viewport-määrittely

Mobiililaitteiden verkkoselaimet pyrkivät pääsääntöisesti skaalaamaan sivun sellaiseksi että se mahtuu kokonaisena päätelaitteen ruudulle. Käyttäjä näkee sivusta yhdellä silmäyksellä kokonaiskuvan ja voi sen jälkeen tarvittaessa skaalata sitä halutun kokoiseksi. Näyttöruudulla hallitaan verkkosivun mittakaavaa, skaalautumista ja monien laitteiden näyttöruudun sommittelua. Jos sivun halutaan skaalautuvan kaikenkokoisille näytöille, skaalautapaa voidaan muuttaa käyttäen HTML-kielen metatiet ominaisuutta.

Viewport-tag sijoitetaan `<head>`-tagin sisään ja se on muotoa:

```
<meta name="viewport" content="attributel, attribute2, ...">
```

Määrite `user-scalable` määrittää sen, voiko käyttäjä zoomata sivua (arvot `yes` tai `no`). Esimerkkikoodissa zoomaus kielletään:

```
<meta name="viewport" content="user-scalable=no">
```

Määritteellä `initial-scale` kerrotaan sivun suurennus sivua avattaessa. Verkkosivu konfiguroidaan skaalautumaan 100 %:iin suhteessa näytön kokoon sivun avautuessa seuraavasti (parametrin arvo 1.0 tarkoittaa 100 %:a):

```
<meta name="viewport" content="initial-scale=1.0,
width=device-width">
```

Edellä parametri `width` määrittelee sivun leveyden ja sille annettu arvo `device-width` tarkoittaa laitteen fyysisen näytön leveyttä. Esimerkiksi, jos laitteen leveys on 320 px, sivu avautuu 320 pikselin levyisenä..

Määritteellä `maximum-scale` kerrotaan suurennuksen maksimiarvo ja `minimum-scale` minimiarvo. Parametrin arvo 1.0 tarkoittaa, että sivua ei suurenneta.

Määritteellä `height` kerrotaan näyttöruudun korkeus. Seuraavalla komennolla verkkosivun korkeus määritellään vastaamaan laitteen fyysisen näytön korkeutta.

```
<meta name="viewport" content="height=device-height">
```

Jos näytön korkeus on 468 px, viewportista tulee 468 px:n korkuinen. Käytännössä tätä määritettä ei kuitenkaan käytetä paljon, sillä näin pystysuunnassa skrollaaminen ei ole enää mahdollista. [1;3.]

Vaikka kuvan ulkoasu muuttuisikin ulkoisesti, varsinainen kuva HTML-dokumentissa pysyy muuttumattomana. Käyttäjät joutuvat siis lataamaan kuvan samalla resoluutiolla ja koolla laitteesta ja näytön koosta riippumatta, mikä johtaa turhaan kaistanleveyden käyttöön ja mahdollisesti hidastaa verkkosivun suorituskykyä. [2.]

4.4 Mediakyselyjen käyttö

Mediakyselyt korjaavat niitä visuaalisia epätäydellisyyksiä, jotka esiintyvät kun näyttöruutu muuttaa muotonsa. CSS3-mediakyselyt antavat suunnittelijan hallita käytettäviä tyylejä laitteen leveyden ja orientaation mukaan. Esimerkiksi resoluutio, leveys, korkeus ja väri voidaan määrittää käytettävän laitteen mukaan. Vaihtoehtoisia sommitteluja siis luodaan laitetypin mukaan, jotta sivut olisivat juuri laitteelle sopivia. [1;3.]

CSS-komennon yleinen muoto on:

```
@media mediatype and|not|only (media feature) {
    CSS-Code;
}
```

Mediakysely koostuu vähintään kahdesta osasta. Kyselyn aloittaa mediatyyppi (media type), esimerkiksi `screen`. Mediatyyppi määrittää kohdelaitteen tyyppin komentamalla käyttäjäagenttia (user agent), kuten selainta, lataamaan määritettyjä tyyliä tietyille mediatyypille. Esimerkiksi `screen`-mediatyyppi käskee selaimen lataamaan tyyliä, jos käytössä on jonkinlainen tietokonenäyttö. `Print`-mediatyyppi puolestaan lataa tyyliä vain tulostettaessa.

Jälkimmäistä osaa sanotaan kyselyksi, esimerkiksi `"min-width: 1024px"` tarkoittaa sitä, että leveydeksi pitäisi olla vähintään 1024 pikseliä. `"Max-width: 1024px"` puolestaan määrittäisi leveydeksi enintään 1024 pikseliä.

Kyselyn osaa `"max-width"` kutsutaan ominaisuudeksi (feature) ja osaa `1024px` arvoksi (value). Esimerkiksi koottu määrittely `screen and (min-width: 1024px)` ilmoittaa sen, että mediatyyppi on näyttö ja selaimen maksimileveys 1024 pikseliä.

Seuraavassa esimerkissä vaihdetaan näytön taustaväri, jos näytön leveys on enintään 300 pikseliä:

```
@media screen and (max-width: 300px) {
    body {
        background-color: lightblue;
    }
}
```

Eri kyselyjä voidaan myös liittää yhteen loogisten operaattoreiden `and`, `or` (pilkku) ja `not` avulla. Useita ominaisuuksia voidaan yhdistää samassa kyselyssä ja luoda näin monimutkaisempia komentoja. Olisi kuitenkin tärkeää tarkistaa, mitä ominaisuuksia eri laitteet tukevat, sillä esimerkiksi iPad 1 ei aluksi tukenut `orientation`-ominaisuutta, kunnes OS-päivitys ilmestyi kaksi kuukautta myöhemmin.

CSS-määrittelyt sisältävät kymmenen mediatyyppiä, jotka on esitetty taulukossa 4. [1;3;4.]

Taulukko 4. Mediakyselytyypit [3.]

TYPE	TARGET DEVICES
all	All devices (default)
braille	Braille tactile feedback devices
embossed	Paged braille printers
handheld	Handheld devices (typically small screen and possibly monochrome)
print	Printing or print preview
projection	Projected presentations
screen	Color computer screen
speech	Speech synthesizers
tty	Media using a fixed-pitch character grid (terminals or teletypes)
tv	Television devices

CSS-määrittelyissä käytettävät piirteet on esitetty taulukossa 5.

Taulukko 5. Mediakyselyn ominaisuudet sekä niiden toiminnot ja käytöt [3.]

FEATURE NAME	DEFINITION	HAS min- AND max- PREFIXES
width	The width of the display area.	✓
height	The height of the display area.	✓
device-width	The width of the device's rendering surface.	✓
device-height	The height of the device's rendering surface.	✓
orientation	Accepts portrait or landscape values.	✗
aspect-ratio	Ratio of the display area's width over its height. For example: on a desktop, you'd be able to query if the browser window is at a 16:9 aspect ratio.	✓
device-aspect-ratio	Ratio of the device's rendering surface width over its height. For example: on a desktop, you'd be able to query if the screen is at a 16:9 aspect ratio.	✓
color	The number of bits per color component of the device. For example, an 8-bit color device would successfully pass a query of (color: 8). Non-color devices should return a value of 0.	✓
color-index	The number of entries in the color lookup table of the output device. For example, @media screen and (min-color-index: 256).	✓

monochrome	Similar to color, the monochrome feature lets us test the number of bits per pixel in a monochrome device.	✓
resolution	Tests the density of the pixels in the device, such as screen and (resolution: 72dpi) or screen and (max-resolution: 300dpi).	✓
scan	For tv-based browsing, measures whether the scanning process is either progressive or scan.	✗
grid	Tests whether the device is a grid-based display, like feature phones with one fixed-width font. Can be expressed simply as (grid).	✗

Mediakyselyjä käytetään niiden visuaalisten epätäydellisyyksien korjaamiseen, jotka saattavat ilmestyä, kun näyttöruutu muuttuu. Mediakysely antaa sivun sisällön esiintyä hienosti luomalla erilaisia asemointeja erilaisille resoluutioille laitteiden tarpeiden mukaisesti. Toisin sanoen tiettyjä tyylejä kohdennetaan tietyille laitteille. [1;2.]

4.5 Keskeytyskohdat

Mediakyselyissä määriteltyjä muutoksia kutsutaan keskeytyskohdiksi (breakpoint). Niissä tyylit muuttuvat laitteen resoluution perusteella. Responsiivisessa toteutuksessa voidaan käyttää yleisiä laiteleveyksiä keskeytyskohtien määrittämiseen. Näitä leveyksiä ovat esimerkiksi 320 pikseliä, 768 pikseliä ja 1024 pikseliä. Ensimmäinen on yksi yleisiä älypuhelimien leveyksiä ja jälkimmäiset kaksi ovat iPadin resoluutioita pysty- sekä vaakasuunnassa.

Toinen tapa on antaa sivun ulkoasun määrittää kohdat, joissa muutokset tapahtuvat. Sivua voidaan pienentää selainikkunassa, jolloin nähdään, koska jokin elementti ei sovi enää ulkoasun mukaisesti näytölle. Tähän kohtaan on hyvä lisätä keskeytyskohta, jotta sivun luettavuus ja ulkoasu säilyvät.

4.6 Suunnittelukehys

Responsiivisen verkkosivun suunnittelu puhtaalta pöydältä on varsin monimutkaista. Pitää valita sopiva määrä sarakkeita, laskea jokaiselle sarakkeella leveys prosentteina, määrittää sopiva keskeytyskohta ja niin edelleen. Suunnittelukehyksen (framework) käyttö helpottaa suuresti ulkoasun suunnittelua ja nopeuttaa verkkosivun tuotantoa merkittävästi. [1;2;4.]

4.6.1 Suunnittelukehyksen käytön edut

Suunnittelukehyksen käyttöä suositellaan, koska suunnittelukehykset säästävät verkkosivun kehittäjien aikaa, varmistavat verkkosivun toimivuuden eri selaimissa, ja niillä on dokumentaatio, yhteisö ja laajennukset eri teemoihin. [2.]

Suunnittelukehyksen käyttö säästää paljon aikaa, sillä suunnittelukehyksellä on jo määritetyt tyylit, kuten pisteverkon leveys, näppäimien tyylit, fonttityylit ja ”CSS reset”, joita tarvitaan rakentamaan verkkosivua. Siten näitä asioita ei tarvitse itse tehdä aluksi, vaan seurataan yksinkertaisesti ohjeistuksia tyylien ja rakenteiden asettamiseen. Esimerkiksi esilatausohjelmalla on pisteverkkopohjan tyylit, perustyyli ja käyttöliittymän tyylit. [2.]

Suosituilla suunnittelukehyksillä on aktiivinen yhteisö, joka laajentaa suunnittelukehyksen toiminnallisuuden. Esimerkiksi Skeletonia on laajennettu Wordpressin teemaan ja Drupaliin. Suunnittelukehyksen käyttöönottamisella ei tarvitse erikseen katsoa, miltä verkkosivu näyttää eri selaimilla, sillä kehityksen kehittäjät ovat jo testanneet sitä eri selaimilla ennen sen julkaisemista. Esimerkiksi Foundation on testattu iOS-, Android-, ja Windows Phone -selaimilla.

Hyvän suunnittelukehyksen kanssa tulee aina dokumentaatio, joka tulee auttamaan tiimityöskentelyssä saamalla jäseniä seuraamaan samaa ohjelmoinnin käytäntöä.

Esimerkiksi Bootstrap ja Foundation ovat tarjonneet yksityiskohtaista dokumentaatiota suunnittelukehyksen käytöstä. [4.]

4.6.2 Sopivan suunnittelukehyksen valinta

suunnittelukehyksiä on monenlaisia. suunnittelukehyksen valinta tehdään verkkosivu-projektin edellytysten mukaisesti. [2.]

Fluid 960 Grid System on helppo oppia, ja sillä saa uusia projekteja nopeasti valmiiksi. Prosenttipohjaiset elementit korvaavat aiemman version 960 Grid Systemin kiinteät (fixed-width) elementit. Blueprint CSS-suunnittelukehys on toinen suosittu kiinteä (fixed-width) CSS-pisteverkkosysteemi (grid system). Pelkästään muutamalla yksinkertaisella CSS:llä sen saa kuitenkin helposti responsiiviseksi. Skeleton, joka on hyvin yksinkertainen suunnittelukehys, on 960 pikseliä leveä ja 16 sarakkeella varustettu. Pisteverkkopohjasta tulee heti responsiivinen integroimalla siihen CSS3-mediakyselyjä. [2.]

Bootstrapilla puolestaan on paljon enemmän ominaisuuksia Skeletoniin verrattuna. Bootstrapia kannattaa käyttää esimerkiksi silloin, kun sivu haluaa tuottaa nopeasti ja muokkauksia halutaan tehdä minimaalisesti. Sillä on tyylikkäästään käyttöliittymäkomponentit, kuten painikkeet ja navigaatio. Lisäksi Bootstrapilla on custom jQuery-liitännäinen, kuten sarkainmerkki, karuselli, "popover" sekä "modal box". Foundationin rakensi kalifornialainen tuotekehitysagentti nimeltä ZURB. Kuten Bootstrap myös Foundation on varustettu määritetyillä tyyleillä, kuten yleisillä käyttöliittymäkomponenteilla, kuten näppäimillä ja navigaatiolla. Sillä on lisäksi jQuery-liitännäinen. Esimerkiksi Pizar ja National Geographin Channel ovat rakentaneet verkkosivunsa käyttäen Bootstrapia. [4.]

4.6.3 Suunnittelukehyksen haittapuolet

Suunnittelukehyksillä on myös haittapuolia. Yleisimmät ongelmat, jotka esiintyvät suunnittelukehyksen käytössä, ovat muun muassa liialliset koodit. Koska suunnittelukehyksiä käytetään laajasti, niillä kuuluisi olla sopiva ulkoasu jokaiselle tilanteelle. Täten ne sisältävät lisätyylejä, joita ei tarvitakaan verkkosivustoon. Vaikka näitä turhia tyylejä on mahdollista poistaa, tämä prosessi voi viedä paljon aikaa ja tuntua ärsyttäväksi.

vältä. Suunnittelukehysten käytön oppiminen, CSS:n tarkistaminen, ID:n, nimien ja HTML:n rakentaminen saattavat myös viedä ison osan suunnittelijan ajasta. Kuitenkin tämä ongelma eliminoiduu ensimmäisen kerran käytön jälkeen, kun suunnittelukehysten käytöstä tulee tuttua.

Suunnittelukehysten käyttö jättää suunnittelijoille vähemmän valinnanvaraa tyylien määrittämisessä. Suunnittelukehyksessä on melkein kaikki asetukset määritetty, mukaan lukien pisteverkon leveys, näppäimien tyylit ja reunasäteet. Näiden määritettyjen asetusten muuttaminen saattaa viedä paljon aikaa ja erehdysten sattuessa pilata koko ohjelmakoodin rakenteen. [4.]

5 Käytettävyys

Käytettävyys on tutkimusala, jonka tarkoituksena on kehittää käyttäjän ja laitteen vuorovaikutusta miellyttävämmäksi ja tehokkaammaksi. Käytettävyttä arvioidaan opittavuuden, tehokkuuden, muistettavuuden, virheettömyyden ja tyytyväisyyden perusteella. Opittavuus kuvaa, kuinka helppoa verkkopalvelun käyttö on ensimmäisellä käyttökeralla. Tehokkuus kertoo, kuinka osaavasti käyttäjä toimii verkkosivulla sen tultua tutuksi. Muistettavuus mittaa sen, miten hyvin käyttäjä muistaa jo oppimansa toiminnot. Virheiden vakavuus ja määrä kuvaavat myös sivun käytettävyttä. [5]

Verkkosivun käytettävyys on hyvä, kun sivun sisältö on loogisesti järjestetty ja selkeä. Oleellisen ja tärkeän sisällön pitäisi näkyä ensimmäisenä. Otsikkojen ja alaotsikkojen lisääminen edistää sisällön selkeyttä. Tekstin tyylin, koon ja värin pitäisi olla tasapainossa. Korostuksia, kursivointia, isoja kirjaimia, alleviivauksia ja lihavoitua ei kannata käyttää liiallisesti. Käytettävyyden kannalta verkkosivun on siis tärkeä olla helppokäyttöinen, helposti opittavissa, miellyttävä ja nopea. Responsiivisen sivun käytettävyttä voi ylläpitää kehittämällä sen navigointia, suurentamalla painikkeet oikean kokoisiksi, välttämällä ahdasta rakennetta sekä soveltamalla Mobile First -ajattelua. [5.]

5.1 Navigointi

Navigoinnin pitäisi löytyä helposti ja nopeasti. Hyvin suunniteltu navigointi auttaa käyttäjiä pääsemään sinne, minne he aikovat ja näyttää heille halutun sisällön. Näytön täyttäminen navigointivalikolla ei ole kuitenkaan paras tapa toteuttaa verkkosivun navigointia, sillä vierailijat eivät muutenkaan käy sivustolla ihaillakseen navigointivalikkoa, vaan katsoakseen sivun asiasisältöä. Vaikka he tulevatkin todennäköisesti hyödyntämään navigointia jossakin vaiheessa, sen ei pitäisi olla ainoa asia, mitä he näkevät.

Olisi kannattavaa poistaa navigointia sivun sisällön tieltä, jotta käyttäjät voisivat nähdä hakemansa informaation. Ratkaisuna on navigointisysteemi, joka näkyy käyttäjälle tarvittaessa ja piiloutuu kun sitä ei tarvita. Alavetovalikko on hyvä vaihtoehto. ”Menu icon” tai ”hint and reveal” ovat sellaisia navigointia, jotka näyttävät linkkilistan, kun käyttäjä painaa itse ikonia. Toisella napsautuksella valikko jälleen kätkeytyy. ”Sticky”-valikkoa

tai "sticky" -otsikkoa olisi parempi välttää, sillä vaikka se näyttääkin hyvältä työpöydällä, se vie paljon tilaa ja rajoittaa käyttäjän kykyä nähdä muuta sivun sisältöä.

Navigointivalikon kutistamista olisi hyvä välttää, sillä on ärsyttävää yrittää koskea pientä navigointilinkkiä sormella. On mahdotonta painaa ihmisen käden sormella linkkiä hiiren osoittimen tarkkuudella, joten lähekkäin olevat linkit saattavat johtaa tahtomatta virheellisiin napsautuksiin. [5.]

5.2 Painikkeiden suunnittelu.

Painikkeiden ja muiden elementtien on oltava tarpeeksi isoja ja eriteltyjä, jottei käyttäjä koskisi tai pyyhkisi elementtiä. Linkin sisällyttäminen alleviivattuun sanaan tai virkkeeseen ei ole enää suositeltavaa, sillä sellaista linkkiä on vaikeampaa napsauttaa mobiililaitteella. [5.]

5.3 Verkkosivun vierittäminen

Verkkosivun kaikkea sisältöä ei tarvitse mahduttaa näyttöruutuun. Käyttäjillä ei ole ongelmaa sivun vierittämisessä, kun he jatkavat hakemansa sisällön katselemista. Mobiililaitteiden käyttäjät ovat jo tottuneet selaamaan verkkosivuja, joten he tietävät jo vaistomaisesti, että lisää sisältöä tulee näkyville vierittämällä. Siten mobiililaitteessa ei ole merkitystä, näkykö sivun koko sisältö heti, kun sivu latautuu. Tärkein sisältö kannattaa kuitenkin panna ensimmäiseksi. [5.]

5.4 Hakutyökalut

Sivustolle tulisi lisätä hakuominaisuudet, varsinkin jos sivulla on paljon asiaa ja sivuja. Haku kannattaa jättää pois vain jos sivuston sisältö on itsestään selvää eikä haulle ole näin tarvetta. Haku näytetään usein tekstilaatikkona, jonka vieressä on painike, jossa lukee sana "etsi", tai vaihtoehtoisesti painikkeessa voi olla suurennuslasin ikoni.

6 Yhteenveto

Nykyisin suurempi osa ihmisistä käyttää internetiä mobiililaitteella tietokoneen sijasta. Sivustoja on erittäin tärkeää rakentaa responsiivisesti, sillä nykyisin erikokoisia laitteita on niin paljon, että eri verkkosivun rakentaminen kullekin laitteelle olisi varsin aikaavievää ja lisäisi kustannuksia. Enemmän henkilöstöä tarvittaisiin luomaan ja ylläpitämään eri verkkosivuja. Erilaisia laitteita on nykyään niin paljon, että yksilöllisen verkkosivun rakentaminen joka laitteelle ei ole enää kannattavaa. Joustavassa verkkosuunnittelussa suunnittelijat voivat luoda sivuja laitteille, joita ei ole vielä keksittykään.

Responsiivisen verkkosuunnittelun perustana ovat joustava sommittelu, joustavat kuvat sekä mediakysely. Kiinteä sommittelu rajoittaa sisällön näkymistä pienillä resoluutioilla. Isommilla resoluutioilla sen sijaan sisältöä ympäröi valkotila. Joustavalla eli prosentteilla luodulla sommittelulla koodatuista sivustoista tehdään responsiivinen siten, että sivusto mukautuu selaimien ja laitteiden kuvakokoon. Pisteverkon käyttö antaa rakennetta ja johdonmukaisuutta.

Näyttöruudun tagilla taas sivu skaalautuu oikein. Mediakyselyiden käytöllä korkeus, leveys ja muut asetukset voidaan määrittää toiveiden mukaisesti. Mediakyselyt luovat sellaista sommittelua, joka on juuri kohdelaitteelle sopiva.

Kuvat voidaan sovittaa joustavalle sommittelulle useilla tavoilla. Helpoin tapa on käyttää koodia `max-width: 100%`, jolla saa kuvan vastaamaan näyttöruudun eli selaimen näkyvän osan leveyttä. Koodi `overflow: hidden` puolestaan peittää leveän kuvan ylimääräisen osan, joka törröttää näyttöruudusta.

Verkkosivun suunnittelussa on tärkeää harkita sivun tulevien vierailijoiden käyttökoke-musta. Monet suunnittelijat rakentavat verkkosivua miettimättä, miten sivun kävijät tulevat kokemaan sivun selaamisen. Verkkosivun pitäisi tarjota mahdollisimman hyvä käyttökokemus millä tahansa laitteella, joka sallii internetin selailun, jotta sivun kävijät saataisiin palaamaan sivulle jälleen uudelleen.

Jokaisen verkkosivun käytön kannattaisi tehdä mahdolliseksi eri laitteilla, sillä tilastojen mukaan ihmiset selailevat nykyään internetiä enemmän mobiililaitteilla kuin tietokoneilla. Tämän kasvun oletetaan jatkuvan tulevaisuudessa. Tämän työn tavoitteena on kan-

nustaa suunnittelijoita rakentamaan responsiivisia verkkosivuja niin, että ne olisivat käytettäviä ja tarjoaisivat loistavan käyttökokemuksen. Esimerkiksi huono navigointi, sivun hidas latautuminen sekä ahtaasti asetetut painikkeet saattavat ajaa vierailijat hyvästelemään kyseisen verkkosivun ja hakeutumaan toisille mieluisimmille sivuille. Verkkosivu pitää suunnitella niin, että se antaa kävijöille loistavan kokemuksen heidän käyttämästä laitteesta riippumatta.

Lähteet

- 1 Ethan Marcotte. 2011. Responsive Web Design. A Book Apart.
- 2 Benjamin LaGrone. 2013. HTML5 and CSS3 responsive web design cookbook. Packt Publishing Ltd.
- 3 Tim Cadlec. 2012. Implementing Responsive Design. New Riders
- 4 Thoriq Firdaus. 2013. Responsive Web Design by Example Beginner's guide. Packt Publishing Ltd.
- 5 Rethinking Usability for Responsive Web Design. 2014. Synecore. Verkkosivu. <<http://engage.synecoretech.com/Portals/141995/docs/responsive-web-design-ebook.pdf>> Luettu 20.8.2014.
- 6 Your Content, Now Mobile. 2012. Content Strategy. Verkkosivu. <<http://alistapart.com/article/your-content-now-mobile>> Luettu 19.8.2014.
- 7 Responsive Web Design: What It Is and How To Use It. 2011. Smashing Magazine. Verkkosivu. <<http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>> Luettu 20.8.2014.
- 8 Responsive Web Design: Missing the Point. 2012. Brad Frost. Verkkosivu. <<http://bradfrostweb.com/blog/web/responsive-web-design-missing-the-point/>> Luettu 20.8.2014.
- 9 Hint and Reveal: Mobile Design Tip to Extend Small Screens. 2012. Mobify. Verkkosivu. <<http://www.mobify.com/blog/mobile-design-tip-hint-reveal/>> Luettu 21.8.2014.
- 10 End User Centric Responsive Web Design. 2012. Jon Arne Saeterås. Verkkosivu. <<https://www.youtube.com/watch?v=32UYNa-Fdgk>> Luettu 21.8.2014.
- 11 Mobile web will overtake the desktop within 5 years. 2011. Verkkosivu. <<http://www.spotzerblog.com/category/mobile/>> Luettu 20.8.2014.
- 12 Mobile First Design: Why It's Great and Why It Sucks. 2013. Verkkosivu. <<http://designshack.net/articles/css/mobilefirst/>> Luettu 19.8.2014.